
IQN2 LLD

Release Notes

Applies to Product Release: 01.00.00.11
Publication Date: July 17th, 2018

Document License

This work is licensed under the Creative Commons Attribution-NoDerivs 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nd/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Contributors to this document

Copyright (C) 2018 Texas Instruments Incorporated - <http://www.ti.com/>



Texas Instruments, Incorporated
5 Chemin des Presses, 4 allée Technopolis
06800 Cagnes sur Mer, FRANCE

Contents

- Overview..... 1
- LLD Dependencies 1
- New/Updated Features and Quality 1
- Resolved Incident Reports (IR) 4
- Known Issues/Limitations 5
- Licensing..... 5
- Delivery Package 5
- Installation Instructions..... 5
- Customer Documentation List 6
- Test Instructions for TCI6630K2L EVM 6

IQN2 LLD version 01.00.00.11

Overview

This document provides the release information for the latest IQN2 Low Level Driver which should be used by drivers and application that interface with IQN2 IP.

IQN2 LLD module includes:

- Compiled library (Big and Little) Endian of IQN2 LLD.
- Source code.
- API reference guide
- Design Documentation

LLD Dependencies

LLD is dependent on following external components delivered in PDK package:

- TCI6630K2L CSL

New/Updated Features and Quality

Release 1.0.0.11

- Updated buildlib.xs to add RULES_MAKE macro to support build based on custom Rules.make

Release 1.0.0.10

- Rename test project name to align with new rules in pdkProjectCreate script
- Fix CSL header include path error to align with Proc-SDK CSL

This is a release tested by the development team on TCI6630K2L EVM.

Release 1.0.0.9

- Fix to be able to set ictl rate control for AID2 and AIL0/1 to zero
- Fix to reset static variables from iqn2 resync procedure when exception counters are reset
- Added support for custom sampling rate in AxC object

- Added exception disabling functions
- Added support for non LTE-or-WCDMA packet size, so regular packet size over 10ms radio frame
- Test utilities
 - Change of the DFE PLL programming to match the PLL user guide recommendations
 - Updated test utilities for generic serdes csl

Release 1.0.0.8

- Fixes in DIO SI to take into account the egress and ingress direction inversion
- Added custom parameter in radio timer object for Wcdma mode to be able to set up TTI ticks bigger than 10ms
- LLD was verified on a secure EVM
- Added support for LTE DIO mode
 - *LTE/WCDMA simultaneous dual mode NOT verified with LTE DIO mode*

Release 1.0.0.7

- Added support for LTE60 (AID2/DFE)

Release 1.0.0.6

- Fixes in iqn2fl for Dio reconfiguration
- Fixes in the Lld for default Tac Dio configuration and Wcdma channel IQ ordering
- Fixes for AID2 CTL channels: RFSDK2 makes use of monolithic descriptors for CTL traffic. (Note: Host packet descriptors for CTL traffic with AID2 works only if the packet descriptors are located in DSP local L2)
- Fixes in the Lld to allow AID2 dual strobe uses cases with 2 LTE groups using the same sampling rate.
- IQN2_resyncProcedure(): fixed the offset calculations. During resync procedure, do not re-enable radio timers, and let the application do it at the appropriate point of time. That avoids a race condition with the enabling of the application AT2 events.

Release 1.0.0.5

- Added support for Sysbios in Lld tests
- Changes in tests for Dfe exception monitoring
- Fix for Coverity issues
- Addition of OSAL functions for RM module
- Added 2 new Iqn2Fl APIs to reconfigure either egress or ingress DIO2 engines (to be used by AZ WCDMA PHY)

- AID2: now using AID2_UAT_EVT_RADT_CMP_CFG to implement egress and ingress delay compensation because ingress axc offset in AID2 is only a 2-bit field, which was not enough for TI LTE PHY

Release 1.0.0.4

- Added APIs for integration with AZ WCDMA PHY
- Added support for LTE40 and LTE80 (AID2/DFE)

Release 1.0.0.3

- Fix in IQN2_resyncProcedure() following testing with RFSDK and external equipment on SoC PHYSYNC pin
- Fix in IQN2_initDio() and DIO parameter initialization

Release 1.0.0.2

- Updated test utilities as Uboot unlocks the kicker and Linux expects the kicker to be unlocked
- Added pre-defined DIO configurations when used with RAC and TAC UMTS accelerators
- Added support for DIO reconfiguration using IQN2 Functional Layer to support RFSDK usecase

Release 1.0.0.1

- Updated DIO engine interface to hold buffer start address for each AxCs in egress and ingress directions
- Verified LTE/WCDMA simultaneous dual mode (DFE BB loopback mode)

Release 1.0.0.0

- Added support for DFE running at 368.64MHz
- Verified LTE10 (DFE) and LTE15 (CPRI/DFE)
- Verified basic functionality of CTL channels (DFE)

Release 1.0.0.0A8

- Removed dependencies on Cppi and Qmss for Yocto build
- Fixes in test utilities for RF integration playback programs

Release 1.0.0.0A7

- Fixes for ARMv7 following Yocto recipe updates
- Added new LLD unit tests for IQN2/DFE

Release 1.0.0.0A6

- Fixes following verification on TCI6630K2L EVM
- Support for IQN2 reset APIs
- Added one API to start BCN timer

- Verified support for external sync on EVM via PHYSYNC
- Verified ARMv7 library (linux user mode LLD): configuration on ARM, runtime on DSP

Release 1.0.0.0A5

- Support for SerDes CSL in test frameworks.
- Support for PSC CSL in test frameworks.
- Support for all IQN2 exceptions.
- Support for LTE1.4, 5, 10 and 20 (CPRI)
- Support for LTE/WCDMA simultaneous dual mode (CPRI)
- Support for AID2/CTL packets
- Support for sync source input to enable the software synchronization process. Verification pending EVM availability
- Alignment (best effort) of the IQN2 LLD APIs with AIF2 LLD APIs: significant rework of the driver code to handle multiple radio standard and multiple radio timer configurations.

Release 1.0.0.0A4

- Support for ARMv7 library (linux user mode LLD)
- Support for AID2 DFE LTE test with configuration on ARMv7 and runtime on DSP

Release 1.0.0.0A3

- Initial ALPHA release of IQN2LLD
- Support for IQN2FL for IQN2 register setup, control commands, and status queries
- Support for IQN2 Driver with the following tested use cases:
 - AIL CPRI (LTE, WCDMA)
 - AIL OBSAI (WCDMA)
 - AID2 DFE (LTE)

Resolved Incident Reports (IR)

Table 1 provides information on IR resolutions incorporated into this release.

Table 1 Resolved IRs for this Release

IR Parent/ Child Number	Severity Level	IR Description
PRSDK-2194	NA	Add RTOS Installer script to autosest SDK_INSTALL_PATH

Known Issues/Limitations

IR Parent/ Child Number	Severity Level	IR Description

Licensing

Please refer to the software Manifest document for the details.

Delivery Package

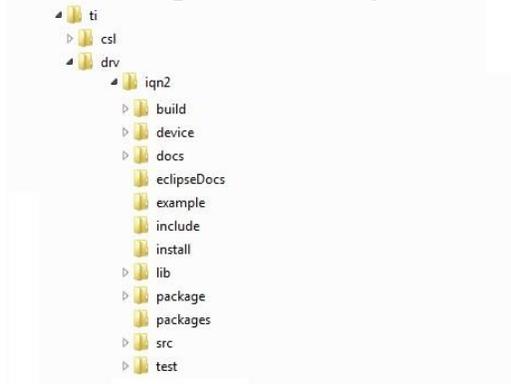
There is no separate delivery package. The IQN2 LLD is being delivered as part of PDK.

Installation Instructions

The LLD is currently bundled as part of Platform Development Kit (PDK). Refer installation instruction to the release notes provided for PDK.

Directory structure

The following is the directory structure after the IQN2 LLD package has been installed:



The following table explains each individual directory:

Directory Name	Description
ti/drv/iqn2	<p>The top level directory contains the following:-</p> <ol style="list-style-type: none"> <u>Environment configuration batch file</u> The file “setupenv.bat” is used to configure the build environment for the IQN2 low level driver. <u>XDC Build and Package files</u> These files (config.bld, package.xdc etc) are the XDC build files which are used to create the IQN2 package. <u>Exported Driver header file</u> Header files which are provided by the IQN2 low level driver

	and should be used by the application developers for driver usage.
ti/drv/iqn2/build	The directory contains internal XDC build related files which are used to create the IQN2 low level driver package.
ti/drv/iqn2/include	The “include” directory has private IQN2 low level driver header files. These files should not be used by application developers.
ti/drv/iqn2/lib	The “lib” folder has pre-built Big and Little Endian libraries for the IQN2 low level driver along with their <i>code/data size information</i> .
ti/drv/iqn2/package	Internal IQN2 low level driver package files.
ti/drv/iqn2/src	Source code for the IQN2 low level driver.
ti/drv/iqn2/test	The “test” directory in the IQN2 low level driver has unit test cases which are used by the development team to test the IQN2 low level driver.
ti/drv/iqn2/eclipseDocs	The “eclipse” directory has files required to integrate IQN2 low level driver documentation with Eclipse IDE’s Help Menu.
ti/drv/iqn2/docs	The directory contains the IQN2 low level driver documentation.

Customer Documentation List

Table 2 lists the documents that are accessible through the /docs folder on the product installation CD or in the delivery package.

Table 2 Product Documentation included with this Release

Document #	Document Title	File Name
1	API documentation (generated by Doxygen)	docs/iqn2lldDocs.chm
2	Design Document	docs/IQN2_LLD_SDS.pdf
3	Software Manifest	docs/IQN2_LLD_SoftwareManifest.pdf

Test Instructions for TCI6630K2L EVM

When MCSDK is installed in the user environment, an Eclipse workspace is created with all PDK DSP examples in <mcsdk_install_path>/pdk_keystone2_3_01_00_01/packages/exampleProjects Alternatively, users can follow instructions from:

http://processors.wiki.ti.com/index.php/MCSDK_UG_Chapter_Developing_PDK#Steps_to_run_example_and.2For_unit_test_projects_on_C66x_Target

○ Switching from CPRI to DFE mode

First step is to verify that the EVM BMC version and FPGA bit file are up-to-date. Early versions don’t support the following. We tested the below with this info from BMC prompt:

BMC Version EVM Version EVM S/N FPGA ver UCD VER

From EVM BMC prompt, change the current bootmode to properly set bit #27:

Example for “dsp no-boot”

CSISC2_0_MUX to 1 → 0x08100001 (set bit no 27) → CSIS2_0 is assigned to AIL (lane 0 and 1)

CSISC2_0_MUX to 0 → 0x00100001 (reset bit no 27) → CSIS2_0 is assigned to DFE/JESD (lane 0 and 1)

BMC command line

BMC> bootmode #15 0x00000000 0x08100001 dsp no-boot → sets mode #15 as DSP no boot mode for CPRI

BMC> bootmode #15 0x00000000 0x00100001 dsp no-boot → sets mode #15 as DSP no boot mode for DFE

○ **Testing on DSP with resource manager server on ARM Linux side:**

Each of the IQN2 DSP tests can be compiled using the “USERM” preprocessing constant to enable support for resource manager, which is required for proper allocation of multicore navigator hardware queues and packet dma channels. When testing IQN2 from DSP side, then ARM Linux needs to first boot and start the resource manager server. Then the IQN2 unit tests are able to allocated hw queues and packet dma channels.

○ **Testing with external phy sync pulse:**

To emulate an external pulse of PHYSYNC pin, the LLD utilities make use of one of the timer64 peripheral to driver timer output #1. The EVM, by default, does not connect timer output #1 to PHYSYNC. To achieve this, a jumper wire is required between pins 4 and 8 on EVM CN5 connector. When observing CN5 label at the top of the EVM, 4 and 8 are located on the right side of the connector (pin 1 is not easy to see):

Default settings are showing in below figure:

